

Віра Оксентюк¹, Ірина Юрчак², Олександр Маркелов³, Максим Шурко⁴,
Богдан Рубаха⁵, Андрій Черниш⁶

¹ Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: vira.m.oksentyuk@lpnu.ua, ORCID 0009-0005-1491-6946

² Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: iryna.yurchak@lpnu.ua, ORCID 0009-0005-9100-8511

³ Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: oleksandr.e.markelov@lpnu.ua, ORCID 0000-0002-2432-0768

⁴ Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: maksym.shurko.pp.2022@lpnu.ua

⁵ Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: bohdan.rubakha.pp.2022@lpnu.ua

⁶ Кафедра автоматизації та комп’ютерно-інтегрованих технологій, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: andrii.chernysh.av.2022@lpnu.ua

АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ CALORIFY НА ОСНОВІ РУШІЯ UNITY

Отримано: вересень 03, 2024 / Переглянуто: вересень 12, 2024 / Прийнято: вересень 30, 2024

© Оксентюк В., Юрчак І., Маркелов О., Шурко М., Рубаха Б., Черниш А., 2024

<https://doi.org/>

Анотація. В сучасному світі в сфері програмного забезпечення все більше місця займають мобільні застосунки для різних галузей. Особливо важливими є мобільні застосунки для підтримання та моніторингу здорового способу життя. Дана робота присвячена продовженню розроблення програмного забезпечення мобільного застосунку Calorify, а зокрема серверної частини. Серверна частина мобільного застосунку Calorify реалізована на основі рушія Unity. мовою програмування C#. C# є об’єктно-орієнтованою строго типізованою МП високого рівня, розробленою і підтримуваною корпорацією Microsoft. В загальному архітектура Calorify побудована на основі компонентно-орієнтованого підходу. Це забезпечує чітке розділення функціональності та легкість масштабування. Всі ключові аспекти додатку, включаючи доповнену реальність AR, гнучкий користувацький інтерфейс та управління даними, реалізовані через модульні компоненти, які відповідають за конкретні задачі. Структура проекту в Unity організовано у вигляді сцен, які містять різні частини додатку, такі як головне меню та AR-функціонал. Така архітектура дозволяє інтегрувати технології для роботи з хмарними сервісами та локальними базами даних, забезпечувати ефективне управління даними та оптимальну взаємодію з користувачем. Особливістю даного мобільного застосунку Calorify є використання математичних, науково обґрунтованих аналітичних формул для розрахунку кількості спожитих калорій та води протягом доби. МЗ Calorify має відповідну кількість баз даних, більшість з яких локальні і створені за допомогою СКБД SQLite в Unity. До віддалених баз даних в проекті можна віднести БД Firebase та БД Image Target Vuforia. МЗ Calorify як основу для авторизації використовує інструменти надані платформою Firebase. Перевагами такого рішення є задовільний спектр та якість послуг аутентифікації користувачів і доступ до віддалених баз даних. Для маніпуляцій даних користувача використовується Firebase Realtime Database. Тому після авторизації, всі введені дані, окрім паролю, зберігаються на віддаленому сервері Firebase.

Ключові слова: мобільний застосунок, архітектура, рушій Unity, база даних, процес авторизації, розрахунок калорій.

Вступ

Підтримання та забезпечення здорового способу життя є важливою стратегією сучасного людства, враховуючи екологічну ситуацію довкілля, технократний розвиток суспільства тощо. Розвиток мікропроцесорної техніки, сучасних смарт технологій, мов програмування суттєво розширює можливості підтримання здорового способу життя за допомогою інтерактивних застосунків мобільних пристроїв (мобільний застосунок - МЗ). Одним з напрямків є розроблення та реалізація мобільних застосунків для менеджменту, моніторингу та підтримання здорового способу життя [1-3]. Застосунки такого спрямування дають можливість рахувати спожиті калорії протягом певного періоду, враховувати фізичні особливості користувача та активність, кількість спожитої води інш. У результаті такі застосунки надають персоналізовані рекомендації з фізичного навантаження, споживання калорій та води тощо. Дана стаття присвячена розробленню МЗ Calorify для підтримання здорового способу життя, який відрізняється від популярних на ринку на даний час [4, 5].

Для проектування, розроблення та реалізації МЗ у сучасному ІТ світі ефективно зарекомендував себе підхід використання спеціалізованих платформ, зокрема рушіїв [1]. На даний час існує багато платформ та рушіїв: Adobe AIR, Android Studio, appMobi, BlackBerry, OpenPlug, Python, Unity тощо та цей перелік щороку розширюється. Кожна з таких платформ має власне інтегроване середовище розробки з відповідними інструментами та засобами для програмування, відлагодження, тестування та впровадження програмного продукту. Відомо, що МЗ в загальному поділяються на кросплатформенні та нативні [2, 3]. Враховуючи переваги та недоліки таких типів застосунків та аналізуючи вимоги до МЗ Calorify для менеджменту, супроводу та моніторингу здорового способу життя для даної розробки вибрано кросплатформенний тип мобільного застосунку [4, 5]. Результати розроблення та реалізація дизайну МЗ Calorify, інтерфейсу користувача та «frontend» частини програмного забезпечення мобільного застосунку Calorify гібридного типу представлено у [4]. Дана робота присвячена розробленню та реалізації «backend» частини програмного забезпечення МЗ Calorify гібридного типу для моніторингу споживаних калорій та підтримання здорового способу життя користувача.

Для розробки МЗ Calorify був обраний ігровий рушій Unity [6-8], який використовується переважно для створення ігор для великої кількості платформ. Однак відомо, що Unity - це потужний і популярний багатоплатформовий рушій для розроблення ігор і додатків для різних платформ, таких як Windows, macOS, Android, iOS і багатьох інших [6-8]. За його допомогою є можливим створення різноманітних інтерактивних проектів для комп'ютерних та мобільних пристроїв, ігрових консолей та інших платформ. Рушій Unity відомий своїм ергономічним та інтуїтивно зрозумілим інтерфейсом, гнучкістю, широким набором інструментів і підтримкою різних технологій, а також підтримкою віртуальної і доповненої реальності.

Оскільки рушій Unity також має різноманітні можливості роботи з графічним інтерфейсом, технологіями штучного інтелекту AI та доповненої реальності AR доцільно використати його для реалізації МЗ Calorify. Основною ж причиною його вибору була саме технологія комп'ютерного зору, яка використовується пакетом Vuforia. Пакет Vuforia надає SDK для двох основних конструкторів МЗ: Android Studio та Unity.

Отже, існує необхідність продовження розроблення та реалізації бекенд частини МЗ Calorify шляхом імплементації сучасних платформ розробки, наприклад Unity. МЗ Calorify в загальному повинен містити усі необхідні компоненти обраного типу архітектури: частину користувача, серверну частину, бази даних, авторизацію та логізацію. Для розрахунку споживання калорій та води доцільно використати математичні аналітичні науково обґрунтовані формули розрахунку.

Постановка проблеми

Метою роботи є розроблення та реалізація бекенд частини програмного забезпечення мобільного застосунку Calorify за допомогою рушія Unity.

Завдання, які необхідно вирішити у даній роботі:

- Аналіз та вибір способу реалізації бекенд частини програмного забезпечення мобільного застосунку Calorify;
- Визначення алгоритму розрахунку спожитих користувачем калорій;
- Реалізація серверної частини мобільного застосунку Calorify за допомогою рушія Unity.

Огляд джерел інформації за тематикою публікації

На даний час існує багато мобільних застосунків для підрахунку спожитих калорій, вироблення рекомендацій для підтримання здорового способу життя, менеджменту активностей тощо. Усі вони реалізовані різними засобами та мають свої переваги та недоліки. Основні відмінності з точки зору дизайну та функціональних можливостей таких МЗ наведено у [4]. У даній роботі варто порівняти популярні платформи для створення МЗ, зокрема рушія Unity [9, 10] та Android Studio [11, 12].

Основні переваги використання Unity полягають у наступному [9, 10]:

- кросплатформність: Unity дозволяє створювати мобільні застосунки для різних платформ, таких як Android і iOS, використовуючи один і той же код.
- графічні можливості: Unity має потужні можливості для роботи з графікою, анімаціями та візуальними ефектами, що особливо важливо для розробки ігор та візуально насичених додатків.
- спільнота і ресурси: Unity має велику та активну спільноту розробників, яка надає безліч ресурсів, таких як уроки, плагіни, шаблони, що полегшує роботу над проектами.

Недоліки використання Unity [9, 10]:

- ресурсомісткість: Unity може бути більш вимогливим до ресурсів пристрою порівняно з іншими інструментами розробки, що може призводити до погіршення продуктивності та тривалого часу реакції.
- Розмір додатку: Зазвичай мобільні додатки, розроблені на Unity, мають більший розмір, оскільки додаткові компоненти Unity додають об'єм до розміру файлів.

Розглянемо Android Studio в контексті порівняння з рушієм Unity. Переваги використання Android Studio [11, 12]:

- нативна розробка: Android Studio дозволяє розробляти мобільні застосунки для платформи Android, використовуючи нативні інструменти та мови програмування.
- оптимізація під Android: додатки, розроблені в Android Studio, можуть бути оптимізовані для працездатності на пристроях з ОС Android, що може покращити продуктивність і ефективність додатків.

Недоліки використання Android Studio для розробки мобільних застосунків:

- Обмежена кросплатформність: Android Studio призначений для розробки лише мобільних застосунків для платформи Android, що обмежує можливості розгортання на інших платформах.
- Порівняно невелика спільнота, яка використовує пакет Vuforia.

Таким чином врахувавши усі переваги та недоліки Android Studio та Unity, для реалізації кросплатформного МЗ Calorify доцільно вибрати рушія Unity.

Існує багато МЗ для підтримання і моніторингу харчування людей. Наприклад [13] розроблено підрахунок калорій за допомогою відповідних таблиць калорійності продуктів для відповідної місцевості. Також враховано вік людей. Але в такому МЗ не має можливості універсального розрахунку для різних вікових категорій. А також є жорстке обмеження на харчові продукти обмеженої географічно території. У [14] наведено архітектуру МЗ для розпізнавання стиглості овочів та фруктів за допомогою вбудованої камери смартфона та машинного навчання. Такі завдання є актуальними і важливими, але недостатньо відкриті математичні основи

розрахунків. Відомо [15], що існує аналітична формула Миффліна — Сан Жеора для розрахунку кількості калорій на добу, яка враховує вагу, зріст, вік та стать людини в стані спокою. Доцільно застосувати науковий аналітичний підхід для надання рекомендацій користувачеві МЗ Calorify щодо споживання кількості калорій протягом доби. Це дозволить науково обґрунтувати спожиту кількість калорій і врахувати ступінь фізичного навантаження і цим самим дозволить отримати бажаний користувачем результат.

Виклад основного матеріалу

В загальному архітектура МЗ складається з наступних основних компонентів: [2, 3]:

1. *Фронтенд* - сторона користувача. Це візуальна частина мобільного застосунку, з якою напряму взаємодіє користувач А саме дизайн програми, інтерактивність та представлення даних. Під час активності користувача здійснюється запит на серверну частину МЗ, де відбувається його обробка та генерується результат. Для МЗ *фронтенд* розробка показана у [4].

2. *Бекенд* - серверна сторона, відповідна частина МЗ, що бере безпосередню участь у обробці даних, отриманих або через користувача, або опосередковано, через запит на стороні фронтенда до бази даних додатка. На основі серверних скриптів здійснюються певні сценарії для обміну даними, підключаються правильні системи та інструменти для управління та зберігання інформації, а також забезпечується загальна продуктивність МЗ.

3. База даних - це впорядкований набір електронних даних із єдиною структурою.

4. Аутентифікація та безпека. За допомогою аутентифікації визначається, як відбувається перевірка особи користувача, який хоче увійти до МЗ. Невід'ємним атрибутом аутентифікації є обліковий запис користувача, що вимагає введення логіна, пароля та, деколи, інших даних. Даний МЗ Calorify має однофакторну автентифікацію.

Архітектура розробленого МЗ Calorify побудована на основі компонентно-орієнтованого підходу, що забезпечує чітке розділення функціональності та легкість масштабування. Всі ключові аспекти додатку, включаючи доповнену реальність (AR), гнучкий користувацький інтерфейс та управління даними, реалізовані через модульні компоненти, які відповідають за конкретні задачі. Така архітектура дозволяє інтегрувати технології для роботи з хмарними сервісами та локальними базами даних, забезпечуючи ефективне управління даними та оптимальну взаємодію з користувачем. Структура проекту в Unity організовано у вигляді сцен, які містять різні частини додатку, такі як головне меню та AR-досвід.

Реалізація на Unity. Приймаючи до уваги переваги та недоліки Android Studio та, для реалізації кросплатформного МЗ Calorify обрано рушій Unity, графічний інтерфейс середовища розробки якого показано на рис.1.

На рисунку позначено: А – меню – дозволяє створювати об'єкти, ігрові сцени, налаштовувати проект і збирати проект; В – симулятор – ілюструє виконання додатку на телефоні з відповідними налаштуваннями; С – кнопки керування проектом – дозволяють запускати, зупиняти і проходитись кроками по симуляції роботи; D – ієрархія – місце, де відображені ігрові об'єкти і їх ієрархія в даній ігровій сцені; Е – проект – провідник файлами і папками проекту; F – інспектор – панель налаштування ігрових об'єктів, спрайтів, скриптів та іншого. Основною мовою програмування станом на 2024 рік є C#. C# є об'єктно-орієнтованою строго типізованою МП високого рівня, розробленою і підтримуваною корпорацією Microsoft. Причини з яких саме вона використовується в Unity:

- простота вивчення: C# має чистий і простий синтаксис, який досить легко засвоїти навіть для початківців. Він базується на подібних до інших мов програмування концепціях, таких як Java та C++, що робить його зрозумілим для багатьох розробників.

- широкі можливості: C# має велику екосистему бібліотек та інструментів, що дозволяє розробникам реалізувати різноманітні функціональність у своїх програмах. Це дозволяє створювати

складні та інноваційні ігри та додатки в Unity.

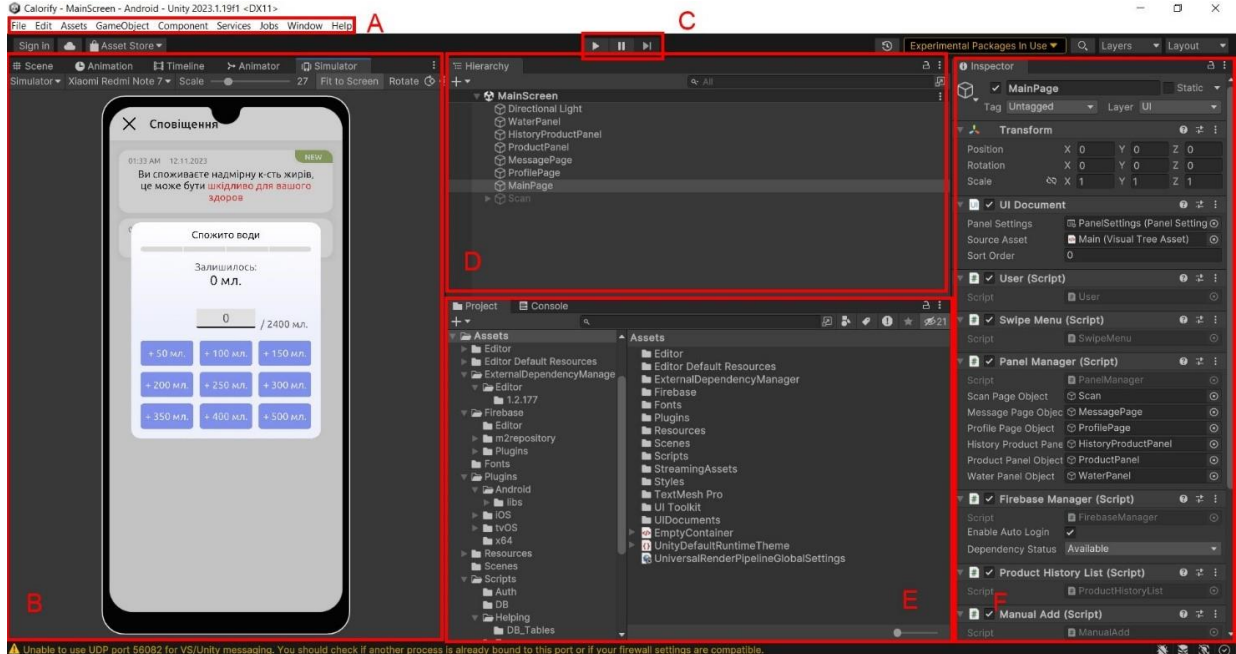


Рис. 1. Графічний інтерфейс середовища розробки:

- підтримка іншими інструментами: C# є однією з основних мов програмування для розробки на платформі .NET, що відкриває доступ до великої екосистеми бібліотек і інструментів для розробки програм поза Unity.

Важливою складовою програмного забезпечення M3 Calorify є доповнена реальність, що реалізована за допомогою вбудованої відеокамери смартфона та у загальному реалізує так званий «сканер продуктів». За допомогою якого є можливість визначення калорійності та основних складових сканованого об'єкту. Функціональна схема сканера M3 представлена на рис. 2.

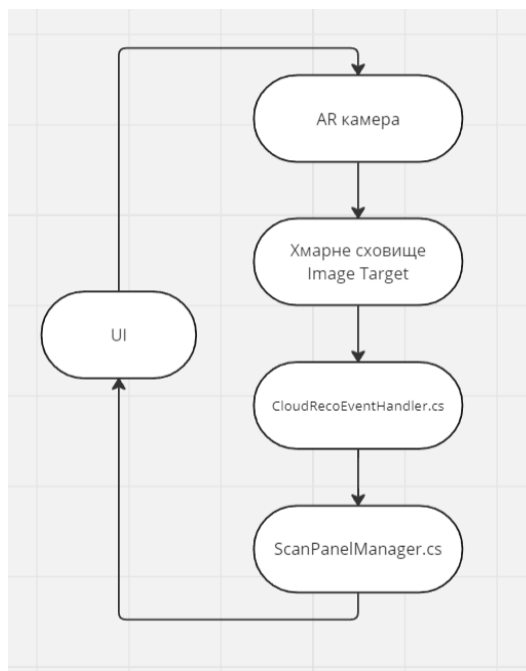
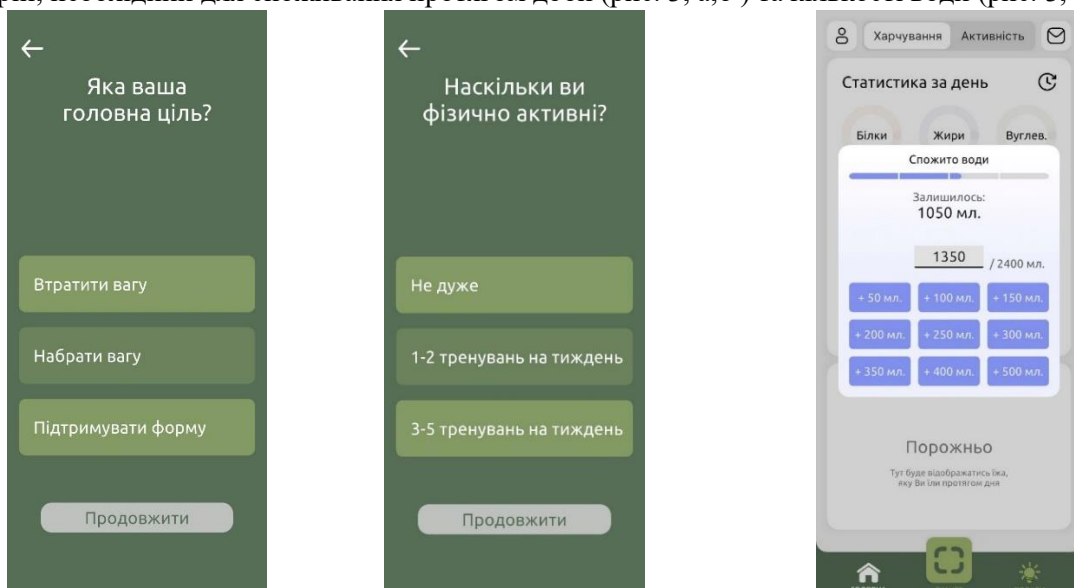


Рис. 2. Графічне відображення модулів для роботи сканера.

Аналітичні формули розрахунку. МЗ Calorify має у своєму складі зокрема дві важливі функціональні можливості для підтримання здорового способу життя: розрахунок кількості калорій, необхідний для споживання протягом доби (рис. 3, а,б) та кількості води (рис. 3, в).



а) мета контролю ваги б) врахування активності в) підрахунок води

Рис. 3. Вигляд вікон з функціями визначення кількості калорій та води

Для ведення здорового способу життя важливо спостереження кількості спожитих калорій протягом доби. Залежно від цієї кількості можливий набір ваги, зниження або підтримання ваги. Існує багато мобільних застосунків із такою функцією [3, 13, 14], проте немає інформації про основу для відповідних розрахунків. В даній роботі у МЗ Calorify запропоновано науково обгрунтований підхід для визначення кількості спожитих калорій протягом доби на основі аналітичних формул для розрахунку на основі [15]. Калорії розраховуються за формулою Міффіна-Сан-Жеора. Для жінок денна норма розраховується за формулою [15]:

$$RMR_w = (mass * 10 + height * 6,25 - age * 5 - 161) * k. \quad (1)$$

Для чоловіків:

$$RMR_w = (mass * 10 + height * 6,25 - age * 5 + 5) * k. \quad (2)$$

де:

RMR – resting metabolic rate (метаболічний ритм у стані спокою);

mass – вага у кілограмах;

height – зріст у сантиметрах;

age – вік;

k – коефіцієнт активності: сидячий спосіб життя — 1.2, легка активність — 1.375, помірна активність — 1.55, висока активність — 1.725, дуже висока активність — 1.9.

З метою скидання ваги, цей результат множиться на 0.9, для підтримання ваги – на 1, а для набору ваги – 1.2.

Отже, для прикладу, розрахуємо денну норму калорій для чоловіка з параметрами: 18 років, зріст – 182 см, маса тіла – 62 кг, помірна активність, мета - набір маси:

$$RMR_M = (62 * 10 + 182 * 6,25 - 18 * 5 + 5) * 1,55 * 1,2 = 3110,85. \quad (3)$$

На основі цих результатів, можемо зробити висновок, що цьому чоловікові для набору маси тіла потрібно споживати близько 3110 ккал/добу.

Інша важлива функція МЗ Calorify полягає у розрахунку і контролі спожитої води (рис. 3, б). Для розрахунку необхідної добової кількості води використовується приблизна формула

[15]:

$$\text{Кількість води (в літрах)} = \text{Вага (в кг)} * 0.03. \quad (4)$$

Згідно з цією формулою, людина зазвичай повинна споживати близько 30 мл води на кілограм ваги щодня. Наприклад, для людини вагою 70 кг це буде близько 2,1 літра води на день. Проте, ця формула є лише загальним орієнтиром, і кожна людина може мати власні індивідуальні потреби у споживанні води.

Бази даних. МЗ Calorify має чималу кількість баз даних (БД), які допомагають користувачеві зручно використовувати застосунок (рис.4). Більшість з них – локальні. Локальні бази даних в Unity створюються за допомогою СКБД SQLite. Це безальтернативний, але від того непоганий варіант збереження даних. До віддалених баз даних в проєкті можна віднести БД Firebase та БД Image Target Vuforia.

Product	
product_id	integer
name	text
type	text
calories	real
prots	real
fats	real
carbs	real
nutri_score	text(1)
vuforia_id	text

а. БД productsdb

History	
id	integer
product_id	integer
mass	real
water_amount	real
date	datetime
user_id	text
time	timestamp

б. БД meals_history

Activity	
activity_id	integer
activity_name	text
activity_type	text
burns_calories	real

в. БД activitiesdb

Activities_history	
id	integer
activity_id	integer
time_spent	real
calories_burnt	real
date	datetime
time	timestamp
user_id	text

г. БД activities_history

Рис. 4. Бази даних мобільного застосунку Calorify.

Процес авторизації. МЗ Calorify як основу для авторизації використовує інструменти надані платформою Firebase (рис. 5). Вона була обрана через низку переваг, що виділяють її на фоні конкурентів. Вирішальною перевагою була безоплатність, однак не менш важливими були й задовільний спектр та якість послуг аутентифікації користувачів і доступ до віддалених баз даних.

Firebase - це платформа розробки мобільних та веб-додатків, яка надає набір інструментів для швидкої розробки, розгортання та управління додатками. Вона включає в себе сервіси для зберігання та синхронізації даних в реальному часі, аутентифікації користувачів, аналітики,

хмарних функцій та хмарного зберігання. Firebase дозволяє розробникам зосередитися на логіці своєї програми, забезпечуючи готові рішення для типових завдань розробки програмного забезпечення та мобільних додатків. Firebase відкриває широкі можливості для швидкого розгортання та масштабування проєктів, сприяючи швидкій розробці та впровадженню рішень.

Для маніпуляцій даних користувача використовується Firebase Realtime Database. Тому після авторизації, всі введені дані, окрім пароллю, зберігаються на віддаленому сервері Firebase. Firebase Realtime Database - це хмарна нереляційна база даних, яка забезпечує синхронізацію та зберігання даних у реальному часі. Вона дозволяє створювати додатки, які можуть взаємодіяти з даними безпосередньо у реальному часі, що робить її ідеальним рішенням для мобільних додатків, онлайн-ігор та інших додатків, які потребують швидкої та надійної синхронізації даних. Вона базується JSON-структурі даних, що дозволяє зберігати дані у вигляді дерева ключ-значення. Вона автоматично синхронізує дані між всіма з'єднаними пристроями в реальному часі, що дозволяє користувачам отримувати оновлення миттєво та без затримок.

```
private IEnumerator CheckAndFixDependenciesAsync()
{
    Task<DependencyStatus> DependencyTask = FirebaseApp.CheckAndFixDependenciesAsync();

    yield return new WaitUntil(() => DependencyTask.IsCompleted);

    dependencyStatus = DependencyTask.Result;
    if (dependencyStatus == DependencyStatus.Available)
    {
        //If they are available Initialize Firebase
        InitializeFirebase();
        yield return new WaitForEndOfFrame();
        if (enableAutoLogin)
            StartCoroutine(CheckForAutoLogin());
    }
    else
    {
        Debug.LogError("Could not resolve all Firebase dependencies: " + dependencyStatus);
    }
}
```

Рис. 5. Функція ініціалізації Firebase.

При вході в застосунок (рис.6,а) спершу ініціалізується сам модуль Firebase, коли все пройдено успішно користувач вибирає метод авторизації. Якщо користувач уже був авторизований, то додаток це прослідкує, підвантажить та відобразить відповідні дані. Розберемо детальніше етап завантаження та відображення даних. Додаток посилає запит на дані користувача до бази даних. Далі додаток зберігає в себе потрібні для подальших операцій дані, в тому числі для відображення графіків та профілю користувача. Усі введені дані при авторизації обробляються на помилку та надсилаються у "frontend" для відображення користувачу.

Отож, розглянемо реєстрацію (рис.6, б). Початковими даними для реєстрації є електронна пошта та пароль. Пошта перевіряється на фактор коректності та попередньої реєстрації. Пароль у свою чергу перевіряється на силу захисту: недостатньо сильний пароль не підійде для реєстрації. Подальші кроки реєстрації дуже прості і працюють за однаковим алгоритмом. Користувач вводить свої дані такі як ім'я, зріст, вага, вік, стать, кількість тренувань на тиждень та мету. Кожен з параметрів перевіряється на відповідності до вимог для реєстрації, у випадку невідповідності видається помилка. Кінцевий етап реєстрації передбачає створення сутності користувача та завантаження інформації у базу даних (рис. 6, в).

Коли користувач вже має зареєстрований акаунт, то він має змогу логінуватись (рис. 7). Логінація відбувається за схожим принципом, що й реєстрація. Спершу емейл проходить перевірку на коректність та на існування в базі даних зареєстрованих користувачів, зрештою

перевіряється пароль та в разі відсутності помилок здійснюється вхід у додаток. Варто зазначити, що пароль, окрім зашифрованої бази даних Firebase, ніде не зберігається, що є необхідною дією для базової безпеки користувача.

```
private IEnumerator AutoLogin()
{
    if (firebaseUser != null)
    {
        if (UnityEngine.SceneManagement.SceneManager.GetActiveScene().name != Scenes.Main.ToString())
        {
            yield return StartCoroutine(SceneLoader.LoadSceneAsync(Scenes.Main));
        }
        SetUserDataWithFirebase();
    }
    else
    {
        if (UnityEngine.SceneManagement.SceneManager.GetActiveScene().name != Scenes.Auth.ToString())
        {
            yield return StartCoroutine(SceneLoader.LoadSceneAsync(Scenes.Auth));
        }
        GetComponent<AuthPanelManager>().OpenSignInPage(null);
    }
}
```

Рис. 7. Функція автологінізації користувача.

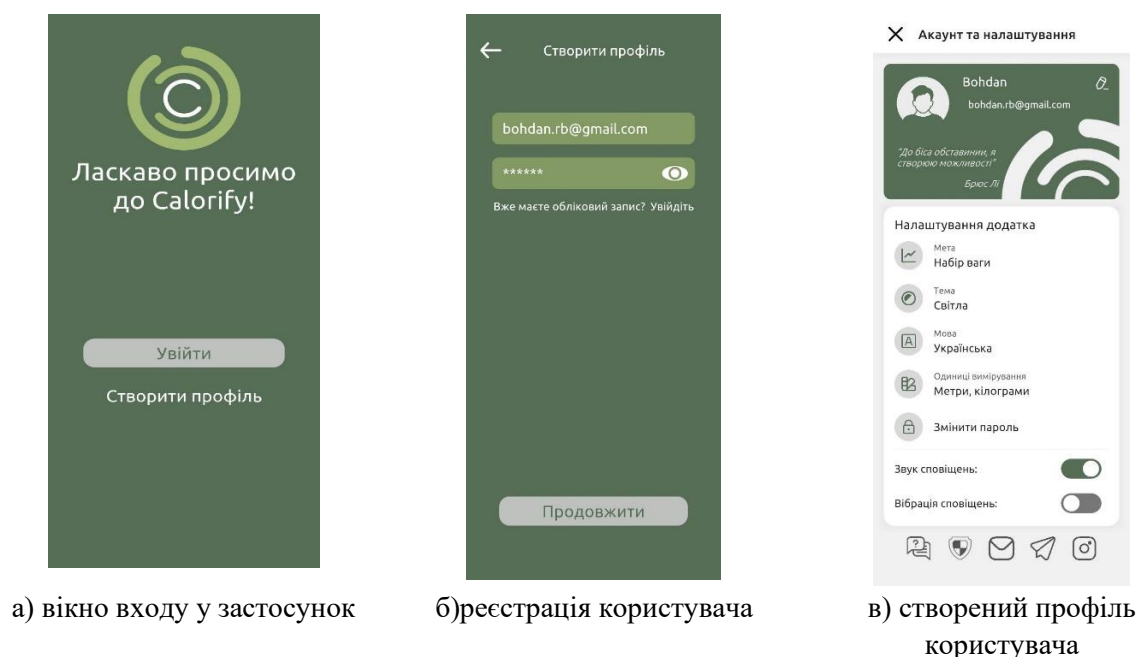


Рис. 6. Вікна реєстрації мобільного застосунку Calorify

Результати та обговорення

Розроблений МЗ Calorify вигідно відрізняється від існуючих використанням аналітичних математичних формул для розрахунку кількості калорій та води. Надано рекомендації для варіантів зниження, підвищення та стабільності ваги.

Архітектура МЗ Calorify містить усі необхідні компоненти, зокрема інтерфейс користувача, серверну частину, бази даних, процес авторизації та афентикації користувача. Визначено та реалізовано архітектуру мобільного застосунку на основі компонентно-орієнтованого підходу, що забезпечує чітке розділення функціональності та легкість масштабування. Всі ключові аспекти

Архітектура та візуалізація мобільного застосунку CALORIFY на основі рушія UNITY

додатку, включаючи доповнену реальність, управління даними та гнучкий користувацький інтерфейс, реалізовані через модульні компоненти, які відповідають за конкретні задачі. Така архітектура дозволяє використовувати технології для роботи з хмарними сервісами та локальними базами даних. Серверна частина МЗ Calorify реалізована на основі рушія Unity. Основні переваги використання Unity полягають у кросплатформності, Unity дозволяє створювати мобільні застосунки для різних платформ, таких як Android і iOS, використовуючи один і той же код. Також графічні можливості Unity можливі для роботи з графікою, анімаціями та візуальними ефектами, що особливо важливо для розробки візуально насичених додатків. Разом з тим, автентифікація – це не єдиний спосіб убезпечити кінцевого користувача вашого мобільного додатка від збоїв роботи та перехвату приватних даних. Зокрема, розробники часто вдаються до реалізації протоколів шифрування даних, впроваджують алгоритми фільтрації вхідних даних, а також забезпечують стабільність системи (наприклад, через тестування навантаження на додаток).

До недоліків розробленого МЗ Calorify можна віднести недостатньо зрозумілу індикацію обраних кнопок та можливість реєстрації лише електронних адрес домену @gmail.com

Потребує подальшого розвитку технологія доповненої реальності за допомогою вбудованої камери смартфона та функціональних можливостей рушія Unity.

Висновки

У даній роботі реалізовано серверну частину МЗ Calorify, призначеного для визначення підрахунку калорій продуктів харчування, на основі рушія Unity. Основні переваги використання Unity полягають у наступному:

кросплатформність для різних платформ, таких як Android і iOS, використовуючи один і той же код.

- графічні можливості для роботи з графікою, анімаціями та візуальними ефектами, що особливо важливо для розробки і візуально насичених додатків.

- спільнота і ресурси, яка надає безліч ресурсів: плагіни, шаблони, що полегшує роботу над проектами

Архітектура МЗ Calorify містить усі необхідні компоненти: інтерфейс користувача (фронтенд), серверну частину, бази даних, реєстрацію користувачів та автентифікацію однорівневу, а також функціонал доповненої реальності. Розроблена та реалізована архітектура МЗ Calorify побудована на основі компонентно-орієнтованого підходу, що забезпечує чітке розділення функціональності та легкість масштабування. Всі ключові аспекти додатку, включаючи доповнену реальність (AR), гнучкий користувацький інтерфейс та управління даними, реалізовані через модульні компоненти, які відповідають за конкретні задачі.

Суттєвою перевагою розробленої серверної частини є розрахунок необхідної кількості калорій за добу на основі математичних, науково обґрунтованих аналітичних формул.

Для подальших досліджень планується удосконалення функціонування частини доповненої реальності за допомогою вбудованої камери смартфона. Також перспективним буде рішення підвищення рівня автентифікації користувачів.

Перелік використаних джерел

[1] Iqbal H. Sarker, Mohammed Moshuiul Hoque, Md. Kafil Uddin, Tawfeeq Alsanoosy, “Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions”, Mobile Networks and Applications, 2021, Volume 26, p.p. 285–303, <https://doi.org/10.1007/s11036-020-01650-z>.

[2] Ichanska N. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки / N. Ichanska, S. Ulko // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2020. – Т. 1 (59). – С. 74-78. – doi:<https://doi.org/10.26906/SUNZ.2020.1.074>.

[3] Anil Patidar; Ugrasen Suman. “Towards Analyzing Mobile App Characteristics for Mobile Software Development”, 2021, IEEE, 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17-19 March, 2021, pp. 786-790.

[4] Ірина Юрчак, Віра Оксентюк, Юлія Гілета, Олег Бабій, Андрій Хіч. "Проектування та реалізація інтерфейсу мобільного застосунку calorify з технологією доповненої реальності", Комп'ютерні системи проектування: теорія і практика, 2024, випуск 6, номер 1, ст. 94-106. <https://doi.org/10.23939/cds2024.01.094>

[5] T. Fatkhulin, R. Alshawi, A. Kulikova, A. Mokin and A. Timofeyeva, "Analysis of Software Tools Allowing the Development of Cross-Platform Applications for Mobile Devices," 2023 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2023, pp. 1-5, doi: 10.1109/IEEECONF56737.2023.10092148.

[6] S. Jangra, G. Singh, A. Mantri, S. Angra and B. Sharma, "Interactivity Development Using Unity 3D Software and C # Programming," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-6, doi: 10.1109/ICCCNT56998.2023.10308030.

[7] Dr. Ashley Godbold, Mastering UI Development with Unity: Develop engaging and immersive user interfaces with Unity , Packt Publishing, 2024; Jiadong Chen; Ed Price, Game Development with Unity for .NET Developers: The ultimate guide to creating games with Unity and Microsoft Game Stack , Packt Publishing, 2022.

[8] Розробка комп'ютерних ігор за допомогою Unity 3D: електронний навчальний посібник для підготовки студентів спеціальності 121 «Інженерія програмного забезпечення» / Укладач: О.М. Ляшенко. – Херсон: видавництво ФОП Вишемирський В.С., 2018. – 220 с

[9] <https://uk.sharpcoderblog.com/blog/category/unity>

[10] Дворецький М. Л., Нездолій Ю. О., Дворецька С. В., Кандиба І. О. Розробка мобільних застосунків для OS Android: навч. посіб. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2021. – 140 с

[11] K. Sudheera, P. Rashmitha and P. S. Maran, "Food Wastage Management Application using Android Studio," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAIC), Salem, India, 2023, pp. 1059-1064, doi: 10.1109/ICAIC56838.2023.10140489.

[12] N. Aukkanit, S. Chopvitayakun, S. Sirichokworakit, J. Chutrtong and K. Kularbphetong, "Mobile Application Development for Required Daily Nutrition Intake Calculation for Thai Elderly," 2023 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2023, pp. 1607-1611, doi: 10.1109/CSCI62032.2023.00266.

[13] D. Fedasyuk, T. Marusenkova and Y. Hura, "Architecture of Mobile Application for Real-Time Calculation of Fruit and Vegetable Mass-to-Volume Ratio," 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, 2023, pp. 1-4, doi: 10.1109/CSIT61576.2023.10324078.

[14] <https://reference.medscape.com/calculator/846/mifflin-st-jeor-equation>.

**Vira Oksentyuk¹, Iryna Yurchak², Oleksandr Markelov³, Maksym Shurko⁴,
Bohdan Rubakha⁵, Andrii Chernysh⁶**

¹ Department of Computer Design Systems , Lviv Polytechnic National University, St. S. Bandery, Lviv, Ukraine, E-mail: vira.m.oksentyuk@lpnu.ua, ORCID 0009-0005-1491-6946

² Department of Computer Design Systems , Lviv Polytechnic National University, St. S. Bandery, Lviv, Ukraine, E-mail: iryna.y.yurchak@lpnu.ua, ORCID 0009-0005-9100-8511

³ Department of Computer Design Systems , Lviv Polytechnic National University, St. S. Bandery, Lviv, Ukraine, E-mail: Oleksandr.E.Markelov@lpnu.ua, ORCID 0000-0002-2432-0768

⁴ Department of Computer Design Systems , Lviv Polytechnic National University, St. S. Bandery, Lviv, Ukraine, E-mail: maksym.shurko.pp.2022@lpnu.ua

⁵ Department of Computer Design Systems , Lviv Polytechnic National University, St. S. Bandery, Lviv, Ukraine, E-mail: bohdan.rubakha.pp.2022@lpnu.ua

⁶ Department of automation and computer-integrated technologies, Lviv Polytechnic National University, St. S. Bandery, Lviv, Ukraine, E-mail: andrii.chernysh.av.2022@lpnu.ua

ARCHITECTURE AND IMPLEMENTATION OF THE CALORIFY MOBILE APPLICATION BASED ON THE UNITY ENGINE

Received: September 03, 2024 / Revised: September 12, 2024 / Accepted: September 30, 2024

© Oksentyuk V., Yurchak I., Markelov O., Shurko M., Rubakha B., Chernysh A., 2024

Abstract. In today's world, mobile applications for various industries are taking up more and more space in the field of software. Mobile applications for maintaining and monitoring a healthy lifestyle are especially important. This work is devoted to the continuation of the software development of the Calorify mobile application, and in particular the server part. The server part of the Calorify mobile application is implemented on the basis of the Unity engine in the C# programming language. C# is an object-oriented, strongly typed, high-level programming language developed and maintained by Microsoft. In general, the architecture of Calorify is built on the basis of a component-oriented approach. This ensures a clear separation of functionality and ease of scaling. All major aspects of the application, including augmented reality AR, flexible user interface and data management, are implemented through modular components that are responsible for specific tasks. The project structure in Unity is organized in the form of scenes that contain different parts of the application, such as the main menu and AR functionality. Such an architecture allows integrating technologies for working with cloud services and local databases, providing effective data management and optimal interaction with the user. The particularity of this Calorify mobile application is the use of mathematical, scientifically based analytical formulas to calculate the number of calories and water consumed during the day. Mobile application Calorify has an appropriate number of databases, most of which are local and created using the SQLite DBMS in Unity. Remote databases in the project include the Firebase database and the Image Target Vuforia database. Mobile application Calorify uses the tools provided by the Firebase platform as a basis for authorization. The advantages of such a solution are a satisfactory range and quality of user authentication services and access to remote databases. Firebase Realtime Database is used for user data manipulation. Therefore, after authorization, all entered data, except for the password, is stored on the remote Firebase server.

Keywords: mobile application, architecture, Unity engine, database, authorization process, calorie calculation.